IN THE CLAIMS

Please amend the claims as follows:

1-7. (Cancelled)

8. (Currently Amended) A processor having a strong ordering instruction architecture comprising:

a store buffer from which a second data value is to be read and stored to a cache memory regardless of whether a first data value that is to be read from the store buffer prior to the second data value being read from the store buffer has been globally observable; and

a global observation store buffer (GoSB) to store only the first and second data values after they have become globally observable, wherein the GoSB comprises an index field for each data value to be stored within the GoSB.

9. (Cancelled)

10. (Previously Presented) The processor of claim 8 further comprising a non-committed store queue (NcSQ) to store data that is stored in the cache memory but has yet to be globally observable.

11. (Original) The processor of Claim 10 wherein the GoSB comprises a count value

corresponding to a number of data stored within the NcSQ.

12. (Currently Amended) The processor of claim 8 wherein the GoSB comprises ~~an index field for each data value to be stored within the GoSB and~~ an address field corresponding to a location a data value to be stored in the GoSB is to be written.

13. (Previously Presented) The processor of Claim 10 wherein the NcSQ comprises an index field to store an index value indicating a location within the GoSB in which a corresponding data value is to be stored after it has become globally observable.

14. (Cancelled)

15. (Original) The processor of Claim 13 wherein the cache memory comprises a line fill buffer (LFB) to store data to be written to a level-1 (L1) cache.

16. (Previously Presented) The processor of Claim 13 wherein data is to be removed from the NcSQ after it has been globally observable.

17. (Original) The processor of Claim 15 wherein the GoSB is to provide a data value to a snooping agent before either the L1 cache or the LSB is to provide the data value to the snooping agent.

18. (Currently Amended) A computer system comprising:

a memory unit to store a first instruction to store a first data value in at least one bus agent and a second instruction to store a second data value in at least one bus agent after the first data value has been stored in at least one bus agent;

a logic to issue the second instruction before the first data has become globally observable;

a level-1 (L1) cache and a line fill buffer (LFB) to store the first and second data values concurrently prior to either of them being detectable by at least one bus agent;

a first bus agent to detect either or both of the first and second data values prior to either or both of the first and second data values being detectable within the L1 cache or the LFB by the first bus agent; and

a storage structure to store the first and second data values after they have become detectable by at least one bus agent, wherein the storage structure comprises an index field for each data value to be stored within the storage structure.

19. (Cancelled)

20. (Original) The computer system of Claim 18 wherein the first bus agent is to snoop either or both of the L1 cache and the LSB for either or both of the first and second data values.

21-22. (Cancelled)

23. (Previously Presented) The computer system of claim 20 wherein either or both of the L1 cache and the LFB are coupled to the first bus agent via a point-to-point bus.

24-25. (Cancelled)

26. (Currently Amended) The computer system of Claim [[25]] 18 wherein the first bus agent comprises an apparatus chosen from a list consisting of a microprocessor, a DRAM, a storage device, and a bus arbitration device.

27. (Currently Amended) An apparatus comprising:

allocation logic to allocate an entry within a global observation store buffer (GoSB) to store globally observable data after a store operation to which the data corresponds becomes non-speculative and before the data is read from a store buffer;

a cache memory coupled to the GoSB to store the data prior to the data becoming globally observable;

read-for-ownership (RFO) logic to obtain exclusive ownership of a line within the cache prior to storing the data within the cache memory, wherein the GoSB comprises an index field for each data value to be stored within the GoSB.

28. (Original) The apparatus of Claim 27 wherein the RFO logic is to obtain exclusive ownership of the line within the cache memory prior to the GoSB entry being allocated.

29. (Original) The apparatus of Claim 28 wherein the store buffer comprises an index to indicate the location of the allocated entry within the GoSB.

30. (Original) The apparatus of claim 29 wherein the allocated entry within the GoSB is to be reallocated if the data becomes globally observable.

31. (Currently Amended) A method for issuing strong ordered memory operations comprising:

issuing a first store operation;

storing a first data associated with the first store operation within a store buffer;

issuing a second store operation;

storing a second data associated with the second store operation within the store buffer;

storing the first data within a storage unit;

storing the second data within the storage unit during a period of time in which the first data is stored within the storage unit and is not globally observable, wherein the storage unit comprises an index field for each data value to be stored within the storage unit.

32. (Original) The method of Claim 31 further comprising issuing a read-for-ownership (RFO) operation to obtain exclusive control over a line within the storage unit prior to one of the first or second data being stored within the storage unit.

33. (Original) The method of Claim 32 further comprising allocating an entry within a global observation store buffer (GoSB) after either the first or second data becomes globally observable.

34. (Original) The method of Claim 33 further comprising updating a counter to reflect a number of data values corresponding to a store address location that are stored within the storage unit but are not globally observable.

35. (Original) The method of Claim 34 wherein the entry is reallocated to store other data after the counter equals zero.

36. (Original) The method of Claim 35 further comprising updating a non-committed store queue (NcSQ) to indicate to the number of data values.

37. (Original) The method of Claim 36 wherein the storage unit is a level-1 (L1) cache.

38. (Original) The method of Claim 36 wherein the storage unit is a line-fill buffer.